Improved Anti-Entropy with Reinforcement Learning

Benjamin Bengfort
University of Maryland
bengfort@cs.umd.edu

Pete Keleher
University of Maryland
keleher@cs.umd.edu

ABSTRACT

Eventual consistency systems can be made more consistent by improving the visibility of a write, that is the time until a write is fully replicated. Gossip based anti-entropy methods scale well but random selection of anti-entropy partners results in less than efficient replication. We propose a simple improvement to pairwise, bilateral anti-entropy; instead of uniform random selection we introduce reinforcement learning mechanisms which assign selection probabilities to replicas most likely to have information. The result is efficient replication, faster visibility, and higher consistency, while still providing high availability and partition tolerance.

INTRODUCTION

A distributed system is made highly available when individual servers are allowed to operate independently without coordination that may be prone to failure or high latency. The independent nature of the server's behavior means that it can immediately respond to client requests, but that it does so from a limited, local perspective which may be inconsistent with another server's response. If individual servers in a system were allowed to remain wholly independent, individual requests from clients to different servers would create a lack of order or predictability, a gradual decline into inconsistency, e.g. the system would experience *entropy*. To combat the effect of entropy while still remaining highly available, servers engage in *anti-entropy sessions* [3] at a routine interval, a process that occurs in the background of client requests.

Anti-entropy sessions synchronize the state between servers ensuring that, at least briefly, the local state is consistent with a portion of the global state of the system. If all servers engage in anti-entropy sessions, the system is able to make some reasonable guarantees about the timeliness of responses; the most famous of which is that in the absence of requests the system will become consistent, eventually. More specifically, inconsistencies in the form of stale reads can be bound by likelihoods that are informed by the latency of anti-entropy sessions and the size of the system [1]. Said another way, overall consistency is improved in an eventually consistent system by decreasing the likelihood of a stale read, which is tuned by improving the *visibility latency* of a write, the speed at which a write is propagated to a significant portion of servers. This idea has led many system designers to decide that "eventual consistency is consistent enough" [2, 4] particularly in a data center context where visibility latency is far below the rate of client requests, leading to practically strong consistency.

More recently there have been two important changes in

SYSTEM DESCRIPTION

A basic sketch of an eventually consistent system is as follows:

PaPoC'18 Worshop Proposal, April 2018, Porto, Portugal
.

BANDIT APPROACHES

EXPERIMENTS

DISCUSSION

CONCLUSION

REFERENCES

[1] Peter Bailis, Shivaram Venkataraman, Michael J. Franklin, Joseph M. Hellerstein, and Ion Stoica. [n. d.]. Quantifying eventual consistency with PBS. 23, 2 ([n. d.]), 279–302. http://link.springer.com/article/10.1007/s00778-013-0330-1

[2] David Bermbach and Stefan Tai. [n. d.]. Eventual consistency: How soon is eventual? An evaluation of Amazon S3's consistency behavior. In *Proceedings of the 6th Workshop on Middleware for Service Oriented Computing* (2011). ACM, 1. http://dl.acm.org/citation.cfm?id=2093186

[3] Douglas B. Terry, Alan J. Demers, Karin Petersen, Mike J. Spreitzer, Marvin M. Theimer, and Brent B. Welch. [n. d.]. Session guarantees for weakly consistent replicated data. In *Parallel and Distributed Information Systems, 1994., Proceedings of the Third International Conference on* (1994). IEEE, 140–149. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=331722

[4] Hiroshi Wada, Alan Fekete, Liang Zhao, Kevin Lee, and Anna Liu. 2011. Data Consistency Properties and the Trade-offs in Commercial Cloud Storage: the Consumers' Perspective.. In *CIDR*, Vol. 11. 134–143.

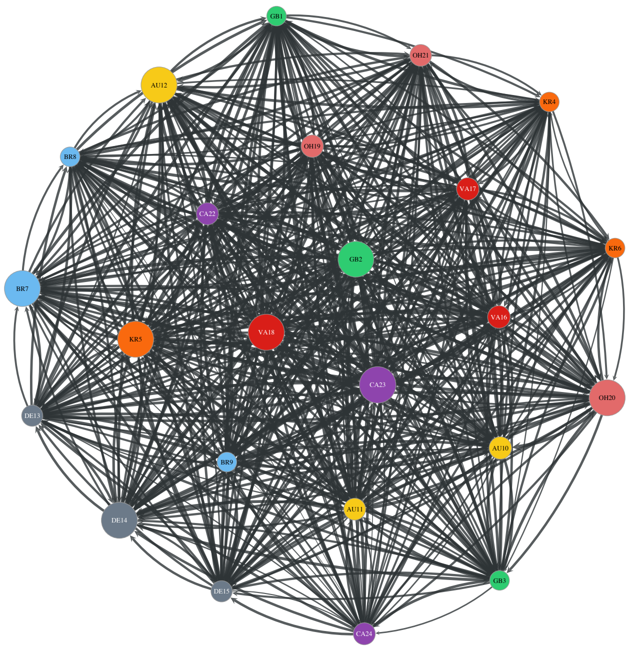

Figure 1: Uniform Selection

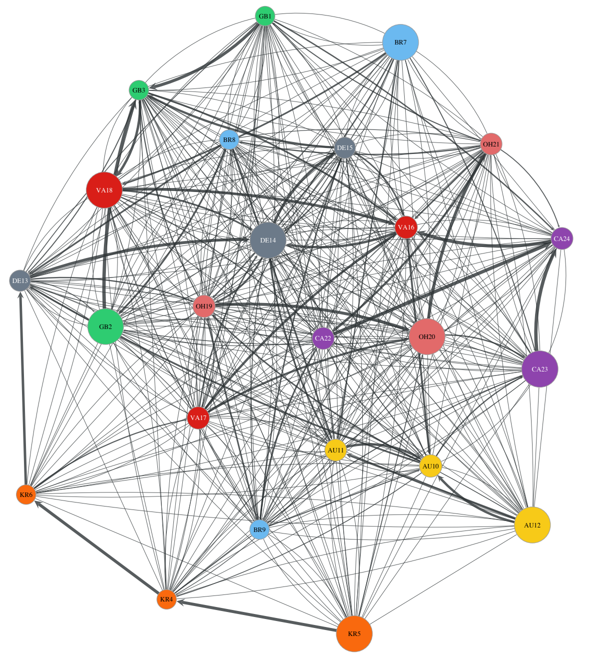

Figure 2: Epsilon Greedy $\epsilon = 0.1$

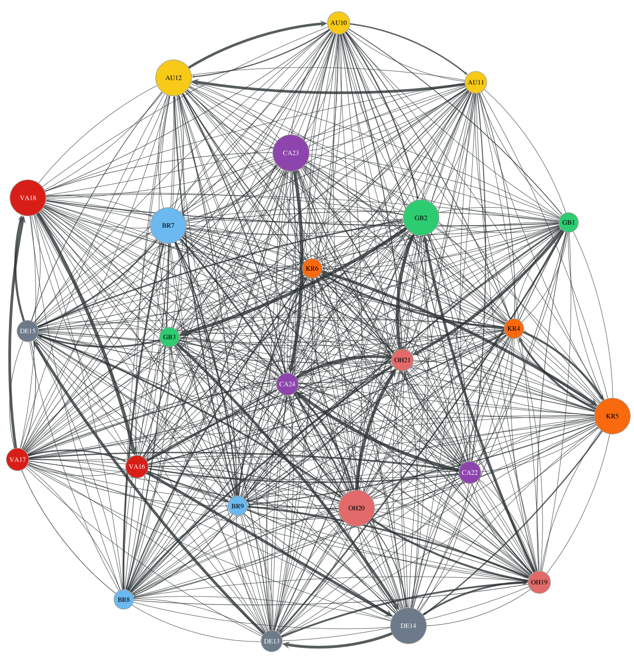

Figure 3: Epsilon Greedy $\epsilon = 0.2$

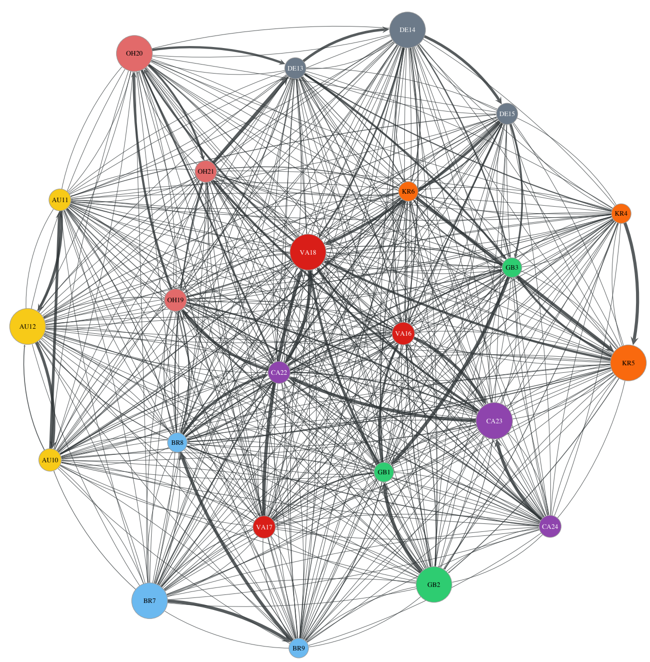

Figure 4: Annealing Epsilon